Graph Trace Regression Estimation

by

Fanwen Zhu

A dissertation submitted to The Johns Hopkins University
in conformity with the requirements for the degree of
Master of Science and Engineering

Baltimore, Maryland
August, 2018

Abstract

While supervised and unsupervised learning on a single graph have been well explored in the literature, supervised learning frameworks on multiple graphs and the pertinent model construction have yet to be well established. In light of the trace regression which has been previously applied to compressed sensing, matrix completion, and multi-task regression, we propose a method to efficiently and accurately estimate a low rank coefficient matrix in the trace regression model where the explanatory variables are the adjacency matrices converted from the graphs. Given a collection of graphs, we utilize the so-called singular value thresholding algorithm that approximates the unknown coefficient matrix in the trace regression model with minimum nuclear norm among all candidates matrices satisfying the designated convex constraints. The algorithm iteratively produces a sequence of matrices $\{\mathbf{X}^k, \tilde{\Theta}^k\}$ where soft-thresholding is operated on the singular values of the coefficient matrix $\tilde{\Theta}^k$. We show through simulation that the singular value thresholding algorithm yields decent prediction accuracy under various specification of the artificial error. Applying the singular value thresholding algorithm on the human brain graphs, we see that it can effectively produces a low-rank estimation of the coefficient matrix while the prediction accuracy is not unacceptable.

Thesis Committee

Primary Readers

Minh Tang (Primary Advisor)
Associate Research Professor
Department of Applied Math and Statistics
Johns Hopkins Whiting School of Engineering

Acknowledgments

First, I want to express my utmost, deepest and sincere gratitude to Professor Minh Tang who has been playing an unique and unparalleled role during my undergraduate and graduate studies in Applied Math and Statistics (AMS) here at Hopkins. I have been knowing Minh for almost three years since the summer of 2015, at which time I wrote my first email to Minh asking about how to prepare for his Applied Statistics and Data Analysis course. Without Minh's passionate and enlightening lectures, I would not have come into the world of data analysis so smoothly and delightfully that I decided to further explore this field by continue studying at Hopkins for a M.S.E. degree. In the fall of 2016, it was Minh who showed immense faith in my academic and teaching ability by designating me as one of the TA for his course. This experience meant so much to me since I was then a senior and never imagined before that I could teach a course mainly enrolled by graduate students. In the fall of 2017, it was again Minh who started guiding me through the entire process of academic research ranging from background reading, coding, scientific thinking and report writing. Whenever I ran into a problem, there was always an answer awaiting me in Minh's office. It is Minh's great patience, amiable personality, and profound knowledge that

keep motivating me to explore and probe further in the realm of science. It is my great fortune to have Minh as my advisor, my mentor and my friend during my time at Hopkins.

I would also like to thank Professor Carey Priebe, my faculty advisor during my sophomore and junior years, and Professor Donniell Fishkind, my faculty advisor during my senior year and Master's program. They are always willing to provide invaluable academic as well as career advice and kind encouragement to me. In addition, I also wish to thank Professor Avanti Athreya for her fascinating lectures on stochastic process in finance and Professor Christopher Sogge in the Math department for his thought-provoking and limpid teaching on real analysis and their willingness to write recommendation letters for my application for the M.S.E. program in the AMS department here at Hopkins.

In addition, I also owe my genuine gratitude to Dr. Ming Teng Koh in the department of Psychological and Brain Sciences who lead me to the wonderland of Standards and Latin dances. Without his effective coaching, I could never have manged to learn eight dances in one year and enjoy the beauty of dancing. Dancing indeed helps me stay joyful during all my Master's program and endows me with energy and power for whatever challenges I encounter. It also gives me tremendous pleasure to thank my dance partner, Yuqi Tan, in the department of Biomedical Engineering. Her enchanting personality, optimistic mind and pursuit of artistic perfection indeed sparks my art appreciation, inspires me to strive to surpass myself, and brings endless happiness to me.

Last but not the least, I must also thank my parents for their decision to send me study abroad at such a prestigious institution, for their constant financial and, more importantly, spiritual support from my childhood, and for, most importantly, giving me birth in this splendid world.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Nowadays, due to its pervasive applications in neuroscience and sociology among others, random graph inference has captured attention in recent literature. While the study of combinatorial graph theory can be dated to as early as 1763 when Leonard Euler published his revolutionary resolution to the conundrum of the bridges of Königsberg , the research on random graph is a relatively young field where the foundational work was accomplished largely by Erdős and Rényi in the late 1950s. In their model, the existence of an edge connecting two vertices are independent of the choice of vertices and are identically distributed Bernoulli random variable with a common probability p. Thereafter, graphs possessing this property are called Erdős-Rényi (or ER) graphs. Since then, two streams of questions are of primary interest: (i) how to better model a random graph and (ii) how to make inference on random graphs. In what follows, we will review the evolution of these two types of questions.

On one side, speaking of the development of the random graph model, the Erdős-Rényi (ER) model – despite its simplicity – enjoys many satisfying

properties (Alon and Spencer (2008), Bollobás, Janson, and Riordan (2007)). However, there are many real life scenarios where the assumption that vertices share a common connection probability fails to hold. For instance, in case of Facebook community where there is a single high dimensional graph in which nodes represent heterogeneous people, it is not sensible to require that it is of the same probability that any two people are friends (so are connected). In addition, the difficulty lies not only in the fact that vertices can be heterogeneous but also in the common situation where the underlying heterogeneous vertex attributes are unobservable. To remedy these issues, Hoff, Raftery, and Handcock (2002) introduced latent space approaches to the study of social network where the objects are graphs with *latent position*. In their setting, while the connection of vertices are still assumed to be independent of the vertex choice, the primary difference is that now each vertex i in the graph is associated with an element z_i of the *latent space* $\mathcal{Z}$ and the probability of connection between vertex i and j, p_{ij}, is no longer uniform across i, j but is determined by a *link* or *kernel* function $\mathcal{K} : \mathcal{Z} \times \mathcal{Z} \to [0, 1]$ such that $p_{ij} = \mathcal{K}(z_i, z_j)$.

In this thesis, we are particularly interested in two variations of the ER models: the *random dot product graph* (RDPG) and the *stochastic block model* (SBM). In RDPG, the latent space is a subspace of Euclidean space $\mathbb{R}^d$ and the $n-$by$-n$ edge connection probability matrix is defined as $\mathbf{P} := \mathbf{Z}\mathbf{Z}^T$ where $\mathbf{Z} \in \mathbb{R}^{n \times d}$ and the link function is given by the inner product of the corresponding rows of $\mathbf{Z}$. Conditional on $\mathbf{P}$, each RDPG has an adjacency matrix $\mathbf{A}$ in which each entry of $\mathbf{A}$ is a realization of a Bernoulli random variable with success probability equal to the corresponding entry of $\mathbf{P}$. In

general, one can generalize the underlying graph to be weighted; this will lead to a weighted version of $\mathbf{A}$ where entries are no longer binary. In addition, there are also cases when the graph is not symmetric, so is $\mathbf{A}$. Nevertheless, for the purpose of simplicity, in this manuscript we will investigate the case when $\mathbf{A}$ is unweighted, symmetric and of no self loop (i.e. diagonal entries of $\mathbf{A}$ are all zero).

On the other side, speaking of the progression of random graph inference, the pioneering work has been devoted into the study of unsupervised learning of a single graph where the goal is to extract important features from a high dimensional graph. Methods such as Adjacency Spectral Embedding (ASE) and Laplacian Eigenmap (LE) which utilize eigen-decomposition of the adjacency matrix $\mathbf{A}$ and the normalized Laplacian matrix $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ respectively are invented and applied to a wide spectrum of problems in social networks and brain connectomics etc. However, there are many other real-world scenarios where the algorithms developed in the aforementioned settings can not account for. For instance, in neuroscience study, often a collection of brain graphs from volunteers are available and it has been shown that certain illness or external stimuli would result in differentiation of brain networks between the healthy and the patients (Bullmore and Sporns (2009)). In this problem, brains regions (resp., the level of underlying neural activity between regions) can be regarded as vertices (resp., edges), and people are interested in locating regions that play the most important role of determining if an individual is sick or not (i.e. the response is binary)(Relión et al. (2017)). Here, the challenge is the classification problem of multiple graphs. For another example, given a

collection of brain graphs, one can also study a series of problems in behavior economics, e.g. how is the brain structure related to an agent's risk preference, educational attainment and income. In this setting, the type of the dependent variable would be ordered categorical or numerical and the problem of interest is to generalize traditional regression in order to incorporate graphs as the input.

In other words, the rising demand for techniques dealing with these problems invites the following two branches of improvement: (i) the extension from one single network to a collection of graphs, and (ii) the evolution from unsupervised learning to supervised regression and classification.

In the past, the problem of graph classification has been substantially studied especially in the context of chemistry and neuroscience. To illustrate, the molecular structure of chemical compounds can be modeled as a graph and Srinivasan et al. (1996) studied how to classify compounds. And in the field of brain networks classification, two ideas of model construction have been explored. The first is to obtain a global summary statistic such as the average path length (Bullmore and Sporns (2009)), based on which a classifier is trained. While it has been shown (Supekar et al. (2008),Liu et al. (2008)) that such global feature can help to diagnosis certain diseases, using the global summary statistic as the input of the model prevents one from interpreting how local differences in the graph contribute to the response. The second approach is to vectorize each adjacency matrix into a row vector and stack those vectors into a numerical matrix. The benefit is that one can apply the many classical high-dimensional classification algorithms such as logistic

regression and decision tree and if variable selection procedure is used in combination, one can even gain interpretation at edge level (J. Richiardi et al. (2011)). Nevertheless, the downside is that vectorization conceals topological structure of the network, suggesting one is unable to identify differentiating communities in the graph.

In this manuscript we restrict our scope to estimating the coefficient matrix Θ in the variations of the so-called graph trace regression model $\mathbf{Y}_i = \text{trace}(\mathbf{X}_i^T\Theta) + \epsilon_i$ where $\mathbf{Y}_i$ is numerical and $\mathbf{X}_i$ is a graph object (e.g. the edge probability matrix $\mathbf{P}_i$ or the adjacency matrix $\mathbf{A}_i$).

This thesis is structured as follows. In Section 2, we introduce the problem setups and discuss trace regression models in general settings. In Section 3, we investigate the algorithm that produces low-rank estimate of the unknown coefficient matrix in trace regression. In Section 4, we specify the trace regression model in a particular setting of interest – when the data matrix represents information of a graph. In Section 5, we illustrate the power of the algorithm of choice by looking at results from simulation and a neuroscience dataset.

References

Alon, N. and J. Spencer (2008). *The Probabilistic Method*. John Wiley, 3 edition.

Bollobás, B., S. Janson, and O. Riordan (2007). "The phase transition in inhomogeneous random graphs". In: *Random Structures and Algorithms* 31, pp. 3–122.

Hoff, Peter D, Adrian E Raftery, and Mark S Handcock (2002). "Latent space approaches to social network analysis". In: *Journal of the American Statistical Association* 97.460, pp. 1090–1098.

Bullmore, E. and O. Sporns (2009). "Complex brain networks: graph theoretical analysis of structural and functional systems". In: *Nature Reviews Neuroscience* 10.3, pp. 186–198.

Relión, Jesús D. Arroyo, Daniel Kessler, Elizaveta Levina, and Stephan F. Taylor (2017). "Network classification with applications to brain connectomics". In: *arXiv preprint at http: //arxiv.org/abs/1701.08140*.

Srinivasan, A., S. H. Muggleton, M. J. Sternberg, and R. D. King (1996). "Theories for mutagenicity: A study in first-order and feature-based induction". In: *Artificial Intelligence* 85.1, pp. 277–299.

Supekar, K., V. Menon, D. Rubin, M. Musen, and M. D. Greicius (2008). "Network analysis of intrinsic functional brain connectivity in alzheimerâĂŹs disease". In: *PLoS Comput Biology* 4.6, e1000100.

Liu, Y., M. Liang, Y. Zhou, Y. He, Y. Hao, M. Song, C. Yu, H. Liu, Z. Liu, and T. Jiang (2008). "Disrupted small-world networks in schizophrenia". In: *Brain* 131, 945âĂŞ961.

J. Richiardi, J., H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville (2011). "Decoding brain states from fMRI connectivity graphs". In: *NeuroImage* 56.2, pp. 616–626.

Chapter 2

Trace Regression Problem

2.1 Notations and models

2.1.1 Notations

We start from establishing the notations. We use boldface (resp., regular) letters to represent vectors and matrices (resp., scalars). As convention, the identity matrix and zero matrix are denoted by $\mathbf{I}, \mathbf{0}$ respectively. We use $\mathbb{R}^{mn}$ (resp., $\mathbb{R}^{m\times n}$) to denote the space of $mn-$dimensional (resp., $m-$by$-n$ dimensional) real vectors (resp., matrices). The Frobenius and nuclear norm of a matrix is denoted by $||\cdot||_F$ and $||\cdot||_*$ respectively. For a matrix $\mathbf{X}$, we use $\mathbf{X}_{ij}$ or $\mathbf{X}_{i,j}$ to denote the $(i,j)^{th}$ entry of $\mathbf{X}$; for a vector $\mathbf{v}$, we use $\mathbf{v}_i$ or $[\mathbf{v}]_i$ to represent the i^{th} component of $\mathbf{v}$. The vectoriezd version of a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ is denoted by $\text{vec}(\mathbf{X}) \in \mathbb{R}^{mn}$ where $\text{vec}(\mathbf{X}) = (\mathbf{X}_1^T, \mathbf{X}_2^T, \ldots, \mathbf{X}_n^T)^T$ in which $\mathbf{X}_i$ represents the i^{th} column of $\mathbf{X}$. We define the inner product of any two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ to be $\langle \mathbf{A}, \mathbf{B}\rangle := \text{trace}(\mathbf{A}^T\mathbf{B})$. In particular, we are interested in the case when $\mathbf{A}^T = \mathbf{A}$, then $\langle \mathbf{A}, \mathbf{B}\rangle = \text{trace}(\mathbf{AB})$.

We let G to represent a *graph* which is an ordered pair of (V, E) where V

(resp., E) is the set of *vertices* (resp., *edges*). We are interested in the case of finite dimensional graphs meaning $|V|$, the cardinality of V, is finite. When $|V| = n < \infty$, we represent V to be $V = \{1, 2, \ldots, n\}$. When there exists an edge connecting vertex $j, k \in V$, then $(j, k) \in E$. In what follows, we let $\mathbf{A}$ denote the so-called adjacency matrix representing a finite graph G. We limit our scope of study to the undirected ($\mathbf{A}_{jk} = \mathbf{A}_{kj}$) and unweighted graphs that contain no self-loops ($\mathbf{A}_{jj} = 0$). In particular, for an unweighted graph, $\mathbf{A}$ is a matrix of only zeros and ones where

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E \end{cases}$$

While these assumptions on $\mathbf{A}$ are not required for the optimization algorithm we use, they match the settings of the applications discussed after the algorithm.

When X is a one-dimensional random variable, $\mathbb{E}(X)$ is the expectation of X as convention. We say the matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a $m-$by$-n$ dimensional random matrix if each element of $\mathbf{X}$ is a one-dimensional random variable and $\mathbb{E}(\mathbf{X})$ represents the element-wise expectation of $\mathbf{X}$.

2.1.2 Trace Regression Models

2.1.2.1 Generic Trace Regression Model

Given a collection of N matrices $\{\mathbf{X}_j \in \mathbb{R}^{m \times n}\}_{i=j}^{N}$ and a set of dependent variable $\{Y_j \in \mathbb{R}\}_{j=1}^{N}$, the generic trace regression model is of the form

$$Y_j = \langle \mathbf{X}_j, \Theta \rangle + \epsilon_i = \text{trace}(\mathbf{X}_j^T \Theta) + \epsilon_i \tag{2.1}$$

where $\Theta \in \mathbb{R}^{m\times n}$ is the underlying true coefficient matrix and ϵ_i's are independent errors with $\mathbb{E}(\epsilon_i|\mathbf{X}_i) = 0$. The goal of this thesis is to estimate the unknown Θ while restricting $\hat{\Theta}$ to be of low rank.

Note that model (2.1) generalizes the traditional linear regression model in a way that $\mathbf{X}_j$'s and Θ in model (2.1) are square diagonal matrices. To illustrate, let $\{\mathbf{X}_j\}_{j=1}^N, \Theta \in \mathbb{R}^{n\times n}$ and define $\mathbf{x}_i := \text{diag}(\mathbf{X}_i) \in \mathbb{R}^n$, $\theta := \text{diag}(\Theta) \in \mathbb{R}^n$ i.e. $\mathbf{x}_i$ (resp., θ) is the vector of diagonal elements of $\mathbf{X}_i$ (resp., Θ). Then model (2.1) is equivalent to $Y_j = \mathbf{x}_j^T\theta + \epsilon_i$ – the classical linear regression model.

Moreover, note that components of Θ in model (2.1) share a similar interpretation with that of θ. In model (2.1), Θ_{ij} is the coefficient corresponding to $\mathbf{X}_{jk}$; that is, Θ_{jk} gauges the contribution of $\mathbf{X}_{jk}$ to the response. In the classical linear regression form, θ_k measures the contribution of the k^{th} component of vector $\mathbf{x}$ (i.e. the k^{th} predictor) to the response.

In this manuscript, we are particularly interested in estimating the unknown Θ with rank constraint. Ideally, we would like to have a low rank estimate $\hat{\Theta}$ to the unknown true Θ. While the sparsity constraint on the unknown coefficient matrix is usually imposed in order to enhance model interpretability, our rank constraint can be regarded as a generalized sparsity constraint. Remarkably, for linear regression model as a special case of the trace model (2.1), a sparse Θ is equivalent to a low rank Θ since Θ is diagonal. In the general form of (2.1), the low rank Θ suggests that the number of effective parameters of (2.1) is small.

2.1.2.2 Generalized Trace Regression Model

Just like in the classical linear regression case where logistic regression is developed to handle the binary response variable, we can also specify model (2.1) particularly when Y_i is dichotomous. Let $\eta_i^\star = \text{trace}(\mathbf{X}_i^T\Theta)$; the logistic trace regression model assumes that

$$\log \frac{\mathbf{P}(Y_i = 1 \Big| \mathbf{X}_i)}{\mathbf{P}(Y_i = 0 \Big| \mathbf{X}_i)} = \text{trace}(\mathbf{X}_i^T\Theta) \tag{2.2}$$

which means given $\mathbf{X}_i$, Y_i follows Bernoulli distribution with success probability equal to $\frac{\exp(\eta_i^\star)}{1+\exp(\eta_i^\star)}$. Note that a special case is when each $\mathbf{X}_i \in \mathbb{R}^{d\times d}$ is a square singleton matrix in a sense that all entries of $\mathbf{X}_i$ are zero except for one nonzero element. Then $\mathbf{X}_i$ can be written as $\mathbf{X}_i = \mathbf{e}_{a(i)}\mathbf{e}_{b(i)}^T$ where $\mathbf{e}_\cdot$ is a column vector of all zeros except for the $\cdot$ entry to be one and $a(i), b(i)$ are just two functions of the index i that produce two integers from 1 to d. Then the logistic regression model can be further rewritten as

$$\log \frac{\mathbf{P}(Y_i = 1 \Big| \mathbf{X}_i)}{\mathbf{P}(Y_i = 0 \Big| \mathbf{X}_i)} = \text{trace}(\mathbf{X}_i^T\Theta) = \Theta_{a(i),b(i)} \tag{2.3}$$

Fan, Gong, and Zhu (2018) discusses how to obtain a low-rank estimation of Θ of model (2.2). In addition to the applications in the compound classification and mental-illness diagnosis mentioned before, model (2.2) can also be utilized to study problems in other fields. See Lee et al. (2014).

References

Fan, Jianqing, Wenyan Gong, and Ziwei Zhu (2018). "Generalized high-dimensional trace regression via nuclear norm regularization". In: *Journal of Econometrics*.

Lee, Eric L, Jing-Kai Lou, Wei-Ming Chen, Yen-Chi Chen, Shou-De Lin, Yen-Sheng Chiang, and Kuan-Ta Chen (2014). "Fairness-aware loan recommendation for micro-finance services". In: *Proceedings of the 2014 International Conference on Social Computing, ACM*, p. 3.

Chapter 3

Optimization Algorithm

In what follows, we will illustrate how we can apply the so-called singular value thresholding algorithm (SVT) (Cai, Candès, and Shen, 2010) to estimate Θ in model (2.1).

3.1 Preliminaries

The SVT algorithm has its name due to its reliance on the *singular value shrinkage operator* discussed below. First recall that for a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ of rank d, its *reduced* singular value decomposition (SVD) can be written as

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad \text{with} \quad \mathbf{\Lambda} = \text{diag}(\{\sigma_i\}_{i=1}^d)$$

where $\mathbf{U}$ (resp., $\mathbf{V}$) is the left (resp., right) singular value matrix of size m by d (resp., n by d) with orthonormal columns and positive σ_i, $\forall i = 1, 2, \ldots, d$. Hereafter, we will work with this reduced form of SVD. Now for $\tau > 0$ – the threshold applied on $\mathbf{\Lambda}$ – we define the singular value soft-thresholding

operator $\mathcal{D}_\tau$ as follows

$$\mathcal{D}_\tau(\mathbf{X}) := \mathbf{U}\mathcal{D}_\tau(\mathbf{\Lambda})\mathbf{V}^T, \quad \text{with } \mathcal{D}_\tau(\mathbf{\Lambda}) := \text{diag}(\{(\sigma_i - \tau)_+\}_{i=1}^d)$$

where for any real number a,

$$(a-\tau)_+ = \begin{cases} a-\tau & \text{if } a \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

Note that although SVD of a matrix may not be unique, the singular value soft-thresholding operator $\mathcal{D}_\tau$ is well defined.

Moreover, for the linear transformation $\mathcal{A}$ defined in (6), we define $\mathcal{A}^*$ to be its adjoint; i.e.

$$\mathcal{A}^*(\mathbf{v}) := \sum_{i=1}^N [\mathbf{v}]_i \cdot \mathbf{P}_i$$

Note that for model (2.3) and (2.4), the definition of $\mathcal{A}^*(\mathbf{Y})$ is the same as above except that $\mathbf{P}_i$ is replaced by $\mathbf{A}_i$ and $\hat{\mathbf{A}}_i$ respectively.

3.2 Algorithm

In the preliminaries, we detailed the derivation of the algorithm for the matrix completion problem. This algorithm can be easily adapted to the trace regression problem as will and can be seen at the end of this section.

From the view of Lagrange multiplier, the optimization problem is

$$\begin{array}{rcl} \min & f_\tau(\mathbf{X}) & \\ \text{subject to} & \mathcal{P}_\Omega(\mathbf{X}) & = \quad \mathcal{P}_\Omega(\Theta) \end{array}$$

where $f_\tau(\mathbf{X}) = \tau||\mathbf{X}||_* + \frac{1}{2}||\mathbf{X}||_F^2$, Θ is the unknown true coefficient matrix in model (2.1), and $\mathcal{P}_\Omega$ is the orthogonal projector onto the span of matrices

vanishing outside of Ω so that

$$[\mathcal{P}_\Omega(\mathbf{X})]_{i,j} = \begin{cases} [\mathbf{X}]_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

The constraint can be rewritten as $\mathcal{P}_\Omega(\Theta - \mathbf{X}) = \mathbf{0}$ so the Lagrangian for this problem is

$$L(\mathbf{X}, \boldsymbol{\lambda}) = f_\tau(\mathbf{X}) + \langle \boldsymbol{\lambda}, \mathcal{P}_\Omega(\Theta - \mathbf{X}) \rangle$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{n_1 \times n_2}$.

Recall that the strong duality result for saddle points suggests that if $(\mathbf{X}^*, \boldsymbol{\lambda}^*)$ is the saddle point of the Lagrangian, i.e.

$$\max_{\boldsymbol{\lambda}} L(\mathbf{X}^*, \boldsymbol{\lambda}) = L(\mathbf{X}^*, \boldsymbol{\lambda}^*) = \min_{\mathbf{X}} L(\mathbf{X}, \boldsymbol{\lambda}^*)$$

then $\mathbf{X}^*$ and $\boldsymbol{\lambda}^*$ are the solutions to the primal and dual problem respectively.

In this light, we can apply Uzawa's algorithm to iteratively approach the saddle point of the problem and therefore obtain the optimal solution. Starting from $\boldsymbol{\lambda}_0 = \mathbf{0}$, the iterative procedure can be defined as

$$\begin{cases} L(\mathbf{X}^k, \boldsymbol{\lambda}^{k-1}) = \min_{\mathbf{X}} L(\mathbf{X}, \boldsymbol{\lambda}^{k-1}) \\ \boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} + s_k \mathcal{P}_\Omega(\Theta - \mathbf{X}^k) \end{cases}$$

where $\{s_k \in \mathbb{R}^+\}_{k \geq 1}$ is a sequence of step sizes. Note that Uzawa's algorithm indeed leads the current iterate in the direction of the gradient or of a subgradient. To see this, let $\bar{\mathbf{X}} := \min_{\mathbf{X}} L(\mathbf{X}, \boldsymbol{\lambda}^{k-1})$, then it follows that

$$\frac{\partial L(\bar{\mathbf{X}}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \frac{\partial f_\tau(\bar{\mathbf{X}})}{\partial \boldsymbol{\lambda}} + \frac{\partial \langle \boldsymbol{\lambda}, \mathcal{P}_\Omega(\Theta - \bar{\mathbf{X}}) \rangle}{\partial \boldsymbol{\lambda}} = \mathcal{P}_\Omega(\Theta - \bar{\mathbf{X}})$$

therefore the update of $\boldsymbol{\lambda}^k$ can be derived by

$$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} + s_k \frac{\partial L(\bar{\mathbf{X}}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \boldsymbol{\lambda}^{k-1} + s_k \mathcal{P}_\Omega(\Theta - \bar{\mathbf{X}})$$

Further, it can be shown that $\bar{\mathbf{X}} = \mathcal{D}_\tau(\mathcal{P}_\Omega(\boldsymbol{\lambda})) = \mathcal{D}_\tau(\boldsymbol{\lambda})$, so the Uzawa's iterates become

$$\begin{cases} \mathbf{X}^k = \mathcal{D}_\tau(\boldsymbol{\lambda}^{k-1}) \\ \boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} + s_k \mathcal{P}_\Omega(\Theta - \mathbf{X}^k) \end{cases}$$

In the context of model (2.1), let $\{\mathbf{M}_i\}$ and $\{\mathbf{Y}_i\}$ be the collection of adjacency matrices and the corresponding response variables, the optimization problem becomes

$$\begin{array}{rl} \min & f_\tau(\mathbf{X}) \\ \text{subject to} & \mathcal{A}(\mathbf{X}) = \mathbf{Y} \end{array}$$

where $\mathcal{A}$ is a linear transformation defined by $[\mathcal{A}(\mathbf{X})]_i = \langle \mathbf{M}_i, \mathbf{X} \rangle$.

The Lagrangian function takes the form of

$$\mathcal{L}(\mathbf{X}, \boldsymbol{\lambda}) = f_\tau(\mathbf{X}) + \langle \boldsymbol{\lambda}, \mathbf{Y} - \mathcal{A}(\mathbf{X}) \rangle$$

where $\boldsymbol{\lambda} \in \mathbb{R}^N$ is the Lagrangian multiplier vector corresponding to the equality constraint $\mathcal{A}(\mathbf{X}) = \mathbf{Y}$. The Uzawa's iteration in this context can be summarized in the following procedure:

Algorithm 1 Singular value thresholding algorithm

procedure SVT($\{\mathbf{M}_i \in \mathbb{R}^{n\times n}, \mathbf{Y} \in \mathbb{R}^n\}_{i=1}^N, \tau > 0, k_{\max}, \epsilon > 0$)
  Set $k \leftarrow 0$, $\boldsymbol{\lambda}_0 \leftarrow \mathbf{0}$, $\mathbf{X}_0 = \mathbf{0}$
  Compute the square of the Lipschitz constant $L^2 = N \times \max\{||\mathbf{M}_i||_F^2\}$
  while $||\mathbf{Y} - \mathcal{A}(\mathbf{X}_0)||_F / ||\mathbf{Y} - \mathcal{A}(\mathbf{X}_k)||_F > \epsilon$ and $k \leq k_{\max}$ **do**
    Compute $\mathcal{A}^*(\boldsymbol{\lambda}_{j-1}) = \sum_{i=1}^N [\boldsymbol{\lambda}_{k-1}]_i \cdot \mathbf{M}_i$
    Compute $\mathbf{X}_k = \mathcal{D}_\tau\big(\mathcal{A}^*(\boldsymbol{\lambda}_{k-1})\big)$
    Compute $\mathcal{A}(\mathbf{X}_k) = \big(\langle \mathbf{M}_1, \mathbf{X}_k\rangle, \ldots, \langle \mathbf{M}_n, \mathbf{X}_k\rangle\big)$
    Choose $\alpha_k \in (0,1)$, set $\delta_k = \alpha_k \cdot \frac{2}{L^2}$ ▷ Choose the step size
    Compute $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_{k-1} + \delta_k(\mathbf{Y} - \mathcal{A}(\mathbf{X}_k))$
    Set $k \leftarrow k + 1$
  return $\mathbf{X}^* = \mathbf{X}_k$ ▷ The estimate is $\hat{\Theta} = \mathbf{X}^*$

where the Lipschitz constant corresponds to the function $\mathcal{F}(\mathbf{X}) = \mathbf{Y} - \mathcal{A}(\mathbf{X})$ in a sense that for any two matrices $\mathbf{X}, \mathbf{X}'$,

$$||\mathcal{F}(\mathbf{X}) - \mathcal{F}(\mathbf{X}')|| \leq L||\mathbf{X} - \mathbf{X}'||_F$$

and X^* is the final estimate of the unknown coefficient matrix Θ.

If the step size δ_k is chosen between $(0, \frac{2}{L^2})$, the SVT algorithm is guaranteed to converge to a unique solution. In model (2.1), $\mathbf{Y}$ is the response variable and $\mathbf{X}$ plays the role of the unknown coefficient matrix Θ.

References

Cai, Jian-Feng, Emmanuel J. Candès, and Zuowei Shen (2010). "A singular value thresholding algorithm for matrix completion". In: *SIAM Journal on Optimization*.

Chapter 4

Graph Models

4.1 Underlying Graph Models

Since we particularly focus on the graph trace regression models, it is important to understand the underlying structure of the graphs. While there are many graph models to choose, two of them are of our interest: (1) random dot product graph (RDPG) and (2) stochastic block model (SBM).

4.1.1 Random Dot Product Graph

Definition. (RDPG). Let F be a $d-$dimensional distribution on a set $\mathcal{Z} \in \mathbb{R}^d$ where $\mathbf{z}_1^T\mathbf{z}_2 \in [0,1]$, $\forall \mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$. Let $\mathbf{Z} = (\mathbf{z}_1^T, \mathbf{z}_2^T, \ldots, \mathbf{z}_n^T) \in \mathcal{Z}^n \subset \mathbb{R}^{n\times d}$ where n corresponds to the number of vertices of the graph. We say $(\mathbf{Z}, \mathbf{A}) \sim RDPG(F)$, if $\mathbf{z}_i$'s are i.i.d. random vectors following $F-$distribution and conditioned on $\mathbf{Z}$, $\mathbf{A}_{jk}$ are independent random variables following Bernoulli distribution,

$$\mathbf{A}_{jk} \sim Bernoulli(\mathbf{z}_j^T\mathbf{z}_k)$$

That said, the random adjacency matrix have the property

$$\Pr\left(\mathbf{A}\middle|\mathbf{Z}\right) = \Pi_{j<k}(\mathbf{z}_j^T\mathbf{z}_k)^{\mathbf{A}_{jk}}(1-\mathbf{z}_j^T\mathbf{z}_k)^{1-\mathbf{A}_{jk}}$$

In other words, rows of $\mathbf{Z}$ are *latent positions* of the adjacency matrix $\mathbf{A}$ to a random dot product graph where rows of Z are independent random vectors following $F-$distribution.

In addition, we define $\mathbf{P} := \mathbf{Z}\mathbf{Z}^T$ as the edge probability matrix; i.e. $\mathbf{P}_{ij}$ is the probability that there is an edge between vertex i and vertex j. The collection of $\mathbf{P}_i$'s is the input in model (4.1). The realizations of these $\mathbf{P}_i$'s – the adjacency matrices $\mathbf{A}_i$'s – are the input of model (4.2). When we treat the latent positions $\mathbf{Z}$ as the parameter, the notation can be rewritten as $\mathbf{A} \sim RDPG(\mathbf{Z})$.

Note that there are two tiers of randomness in the RDPG model. The underlying randomness lies in the latent position $\mathbf{Z}$ (rows of $\mathbf{Z}$ are random vectors) which makes parameter of the *Bernoulli* distribution –the inner product of two rows of $\mathbf{Z}$– for each $\mathbf{A}_{jk}$ a random variable. The higher tier of randomness is in the edge probability matrix $\mathbf{P}$ where each entry in its realization $\mathbf{A}$ follows a *Bernoulli* distribution.

4.1.2 Stochastic Block Model

The stochastic block model (SBM) was introduced in the manuscript by Holland, Laskey, and Leinhardt (1983). It is a special case of independent-edge random graph in a sense that the vertices can be grouped into K *communities* (or *blocks*) and the probability that two nodes are connected is determined by the block membership of the nodes. That said, SBM is usually

characterized by the following two features: (1) an edge probability matrix $\mathbf{B} \in [0,1)^{K\times K}$ whose (i,j)th entry indicates the probability that a node in block i is connected to a node in block j, and (2) a block assignment function $\nu : \{1,2,\ldots,n\} \mapsto \{1,2,\ldots,K\}$ that manifests the block membership of each vertex. In other words, now the edge probability matrix $\mathbf{P}$ has the following property

$$\mathbf{P}_{i,j} = \mathbf{B}_{\nu(i),\nu(j)}$$

and the notation is $\mathbf{A} \sim \text{SBM}(\mathbf{B}, \nu)$.

In what follows, We present an alternative characterization of SBM as a special case of RDPG model:

Definition. (SBM). Given $\mathbf{A} \in \mathbb{R}^{n\times n} \sim RDPG(\mathbf{Z})$, we say $\mathbf{A}$ is an adjacency matrix of a $K-$block SBM graph if there are K distinct rows in $\mathbf{Z} = (\mathbf{z}_1^T, \ldots, \mathbf{z}_k^T)$ where $\mathbf{z}_i^T$ is the i^{th} row of $\mathbf{Z}$. Further, we define the block assignment function $\{1,2,\ldots,n\} \mapsto \{1,2,\ldots,K\}$ such that $\nu(j) = \nu(k)$ if and only if $\mathbf{z}_i^T = \mathbf{z}_j^T$. We then write

$$\mathbf{A} \sim \text{SBM}(\nu, \{\mathbf{z}_i\}_{i=1}^K)$$

Furthermore, we consider the case when the block membership is not predetermined; i.e. each vertex is randomly assigned to a block. To be specific, let $\pi \in (0,1)^K$ such that $\sum_{k=1}^n \pi_k = 1$ and assume that the block assignments $\nu(1), \ldots, \nu(n)$ are i.i.d. random variables following Categorical(π) In other words, $\Pr(\nu(i) = j) = \pi_j$ for $j = 1,2,\ldots,K$. Now we write

$$\mathbf{A} \sim SBM(\pi, \{\mathbf{z}_i\}_{i=1}^K)$$

4.2 Graph Trace Regression Settings

In this thesis, we focus on the case when the data are a collection of graphs $\{G_i = (V_i, E_i)\}_{i=1}^N$. For $\mathbf{X}_i$, there are three possible cases: (i) $\mathbf{X}_i = \mathbf{P}_i \in \mathbb{R}^{m\times m}$ where $\mathbf{P}_{jk} = \Pr\big((j,k) \in E_i\big)$; that is, the $(j,k)^{th}$ entry of $\mathbf{P}_i$ is the probability that there is an edge connecting vertex i and vertex j, (ii) $\mathbf{X}_i = \mathbf{A}_i \in \mathbb{R}^{m\times m}$; i.e., the i^{th} sample point is the adjacency matrix of G_i, and (iii) $\mathbf{X}_i = \hat{\mathbf{A}}_i$ where $\hat{\mathbf{A}}_i$ is a low rank approximation for $\mathbf{A}_i$. In what follows, we discuss the rationales and characteristics of model (2.1) in the three types of $\{\mathbf{X}_i\}_{i=1}^N$.

(i) $\mathbf{X}_i = \mathbf{P}_i \in \mathbb{R}^{m\times m}$.

Since we assume G_i's are undirected graphs, $\mathbf{P}_i$'s are thus symmetric. This implies model (2.1) becomes

$$Y_i = \langle \mathbf{P}_i, \Theta\rangle + \epsilon_i = \text{trace}(\mathbf{P}_i^T\Theta) + \epsilon_i = \text{trace}(\mathbf{P}_i\Theta) + \epsilon_i \tag{4.1}$$

where $\Theta \in \mathbb{R}^{m\times m}$ is the true coefficient matrix and ϵ_i are independent errors with $\mathbb{E}(\epsilon_i|\mathbf{P}_i) = 0$. Note that because $\mathbf{P}_i$ encapsulates the underlying true graph structure information, there is no measurement error in the input $\{\mathbf{P}_i\}_{i=1}^N$. However, in applications, we are usually only able to see the realizations of $\mathbf{P}_i$'s, i.e. the G_i's and the corresponding $\mathbf{A}_i$'s, but unable to observe $\mathbf{P}_i$'s. This leads to the following second variation of model (2.1).

(ii) $\mathbf{X}_i = \mathbf{A}_i \in \mathbb{R}^{m\times m}$.

Again, as G_i's are undirected, $\mathbf{A}_i$'s are also symmetric, suggesting model

(2.1) becomes

$$Y_i = \langle \mathbf{A}_i, \Theta \rangle + \epsilon_i = \text{trace}(\mathbf{A}_i^T \Theta) + \epsilon_i = \text{trace}(\mathbf{A}_i \Theta) + \epsilon_i \tag{4.2}$$

where $\Theta \in \mathbb{R}^{m \times m}$ is the true coefficient matrix and ϵ_i are independent errors with $\mathbb{E}(\epsilon_i | \mathbf{P}_i) = 0$. We posit that there is measurement error in the input $\{\mathbf{A}_i\}_{i=1}^N$ as it is the case in many applications.

In this regard, model (4.2) can be viewed as noisy version of model (4.1) where the noise not only lies in ϵ_i but in the input $\mathbf{A}_i$ as well. While there are assorted ways of denosing a matrix, in this manuscript we replace $\mathbf{A}_i$ by its low rank approximation $\hat{\mathbf{A}}_i$ in model (4.2). This leads to the third variation of model(2.1) as follows.

(iii) $\mathbf{X}_i = \hat{\mathbf{A}}_i \in \mathbb{R}^{m \times m}$.

The particular low rank approximation technique we adopt is based on the spectral decomposition of $\mathbf{A}_i$. Since $\mathbf{A}_i$ is a real symmetric matrix, then

$$\mathbf{A}_i = \mathbf{U}_i \mathbf{D}_i \mathbf{U}_i^{-1} = \mathbf{U}_i \mathbf{D}_i \mathbf{U}_i^T \quad \text{with} \quad \mathbf{U}^{-1} = \mathbf{U}^T$$

where $\mathbf{D}_i = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \in \mathbb{R}^m$. Suppose we choose the first d eigenvalues of $\mathbf{A}_i$ to construct the low rank approximation, then

$$\hat{\mathbf{A}}_i = \mathbf{U}_i \mathbf{D}_i^{(d)} \mathbf{U}_i^T \quad \text{with} \quad \mathbf{D}_i^{(d)} = \text{diag}\{\lambda_1, \ldots, \lambda_d, 0, \ldots, 0\} \in \mathbb{R}^{m \times m}.$$

Clearly, $\hat{\mathbf{A}}_i$ is also symmetric.

Now it follows that model (4.2) becomes

$$Y_i = \langle \hat{\mathbf{A}}_i, \Theta \rangle + \epsilon_i = \text{trace}(\hat{\mathbf{A}}_i^T \Theta) + \epsilon_i = \text{trace}(\hat{\mathbf{A}}_i \Theta) + \epsilon_i \tag{4.3}$$

where $\Theta \in \mathbb{R}^{m \times m}$ is the true coefficient matrix and ϵ_i are independent errors with $\mathbb{E}(\epsilon_i | \mathbf{P}_i) = 0$.

References

Holland, P. W., K. Laskey, and S. Leinhardt (1983). "Stochastic blockmodels: First steps." In: *Social Networks* 5, pp. 109–137.

Chapter 5

Experiments

In this chapter, we present results from a simulation study and a neuroscience application to illustrate the applicability of the SVD algorithm on the problem of trace regression estimation.

5.1 Simulation

In this first experiment, we show that the MSE of the testing sample can be as low as $Var(\epsilon_i)$ in model (4.1), therefore suggesting SVT yields decent prediction accuracy while keeping the estimator to be of low rank.

In this case, we simulate 600 sample points $\{(G_i \in \mathbb{R}^{200\times 200}, \mathbf{Y}_i \in \mathbb{R})\}_{i=1}^{600}$ representing 600 individuals with G_i being the graph and $\mathbf{Y}_i$ being the dependent variable. We then split the data set into training and testing sets which contain 500 and 100 sample points respectively. Further, we categorize sample points into four classes: male and young (M-Y), male and old (M-O), female and young (F-Y), and female and old (F-O). In the training (resp., testing) set, each subcategory contains 25 (resp., 125) sample points.

Block	Young	Old
1	1-100	1-110
2	101-110	111-120
3	111-200	121-200

Table 5.1: Block assignment

We add complexity by assuming that $G_i \sim SBM(\mathbf{B})$ while different age (resp., gender) groups have different block assignments (resp., edge probability matrix). To be specific, for each subclass, we first simulate the RDPG edge probability matrix $\mathbf{P}_i = \mathbf{Z}_i\mathbf{Z}_i^T$ where $\mathbf{Z}_i \in \mathbb{R}^{200\times 3}$ and rows of $\mathbf{Z}_i$ are i.i.d. Dirichlet random vector of size 3.

Next, we specify the block assignment as in Table 5.1.

Note that we artificially make block 2 to be the special block containing vertices $\{V_i\}_{i=101}^{120}$. Its particularity lies in its within-block edge connection probability: in the edge probability matrix of either man or woman, $\mathbf{B}_{2,2}$ is considerably larger than other entries of $\mathbf{B}$. This configuration of block assignment and the simulation of $\mathbf{B}$ mean that there are, in expectation, more edges connecting vertices that are both in block 2.

Based on the block assignment, we then convert each $\mathbf{P}_i \in \mathbb{R}^{200\times 200}$ into $\mathbf{B}_i \in \mathbb{R}^{3\times 3}$ where $[\mathbf{B}_i]_{j,k}$ (i.e. the $(j,k)^{th}$ entry of matrix $\mathbf{B}_i$ where $j,k \in \{1,2,3\}$) is the mean of those entries of $\mathbf{P}_i$ which represent the probability that a vertex in block j is connected to a vertex in block k. Each G_i is a realization of $SBM(\mathbf{B}_i)$.

According to the choice of parameters in the Dirichlet distribution, within each gender group (male and female), $\mathbf{B}_i$'s are alike in a sense that $[\mathbf{B}_i]_{j,k}$

should concentrate around the value equal to the mean of $\{[\mathbf{B}_i]_{j,k}\}_{i=1}^{200}$ for $j, k \in \{1, 2, 3\}$. Therefore the element-wise mean of $\mathbf{B}_i$'s in each gender group can summarize the information about edge connectivity probability:

$$\mathbf{B}^{(\text{Male})} = \begin{bmatrix} 0.43 & 0.27 & 0.27 \\ 0.27 & 0.8 & 0.27 \\ 0.27 & 0.27 & 0.43 \end{bmatrix}, \qquad \mathbf{B}^{(\text{Female})} = \begin{bmatrix} 0.43 & 0.27 & 0.27 \\ 0.27 & 0.75 & 0.27 \\ 0.27 & 0.27 & 0.43 \end{bmatrix}$$

Note that the only remarkable difference lies in the $(2, 2)^{th}$ entry of $\mathbf{B}$. This means on expectation, men have more edges connecting vertices that are both from block 2 than women do.

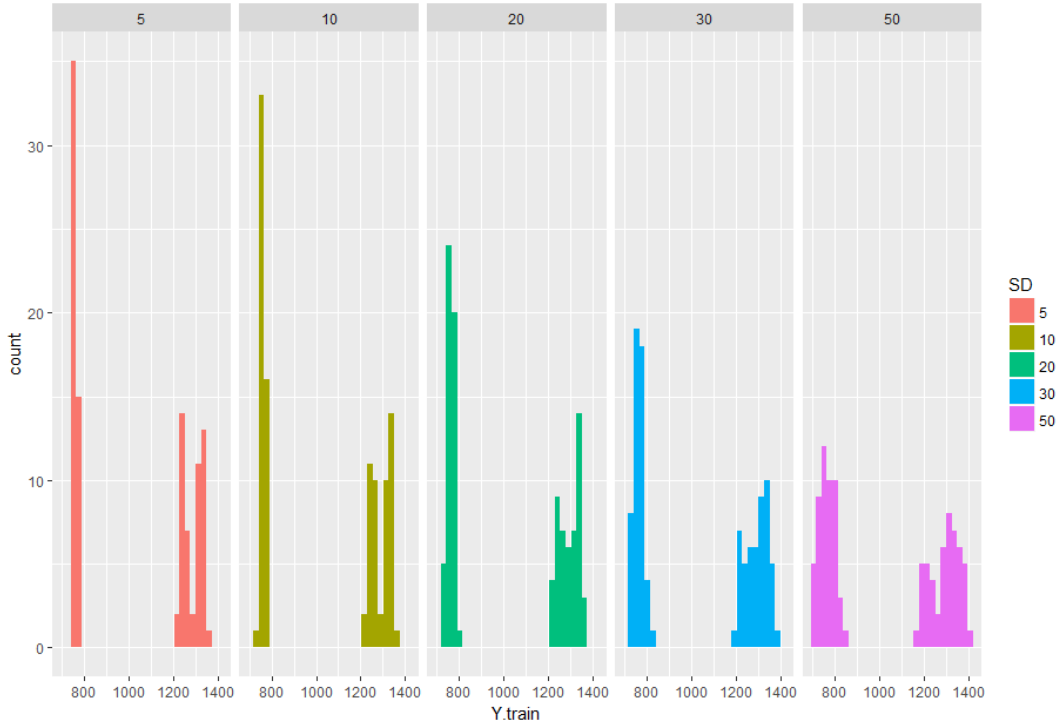
We then simulate the underlying true coefficient matrix Θ with the restriction that all entries are 0 except those in the position (upper left) $\{101, \ldots, 110\} \times \{101, \ldots, 110\}$ and (bottom right) $\{110, \ldots, 120\} \times \{110, \ldots, 120\}$ where $\times$ is the Cartesian product. Note that the upper left and bottom right positions correspond to block 2 of the young and the old respectively. Furthermore, the upper left entries are drawn independently from $Unif(16 - \delta, 16 + \delta)$ and the bottom right from $Unif(1 - \delta, 1 + \delta)$ where $\delta = 0.1$ in both cases. Together, these imply that (i) only those edges which connect vertices in block 2 will contribute to the response and (ii) relatively speaking, contribution of block 2 is greater in the young group than in the old one. So far we have set up G_i and Θ; then we can simulate the response variable $\mathbf{Y}$ using the formula

$$\mathbf{Y}_i = \text{trace}(\hat{\mathbf{A}}_i \Theta) + \epsilon_i$$

where we try out five different settings of ϵ_i where $\epsilon_i \overset{\text{i.i.d.}}{\sim} Normal(0, SD^2(\epsilon_i))$ and $SD(\epsilon_i) \in \{5, 10, 20, 30, 50\}$. The simulation can be seen below where the two clumps correspond to the difference in the parameters of the Uniform

distribution:

Figure 5.1: Distribution of training Y



Next, we train $\mathbf{Y}_i^{(\text{train})}$ by $\hat{\mathbf{A}}_i^{(\text{train})}$ using SVT and then fit $\mathbf{Y}_i^{(\text{test})}$. The output is summarized in Figure 5.3.

From the simulation result table, we see that (i) MSE of both the training sample and the testing sample are substantially smaller than $Var(\mathbf{Y}_i^{(\text{train})})$ and $Var(\mathbf{Y}_i^{(\text{test})})$, suggesting SVT gives decent prediction accuracy and (ii) there is, to some degree, overfitting as can be seen from the relatively larger scale of $Var(\mathbf{Y}_i^{(\text{test})})$ compared to $Var(\mathbf{Y}_i^{(\text{train})})$. In addition, despite depending on the choice of threshold value τ, we can generally obtain a low rank estimate $\hat{\Theta}$ where $rank(\Theta) = 20$ and $\Theta \in \mathbb{R}^{200\times 200}$. That is to say, SVT achieves our two ideal goals: sound prediction accuracy and low rank.

Figure 5.2: Distribution of testing Y

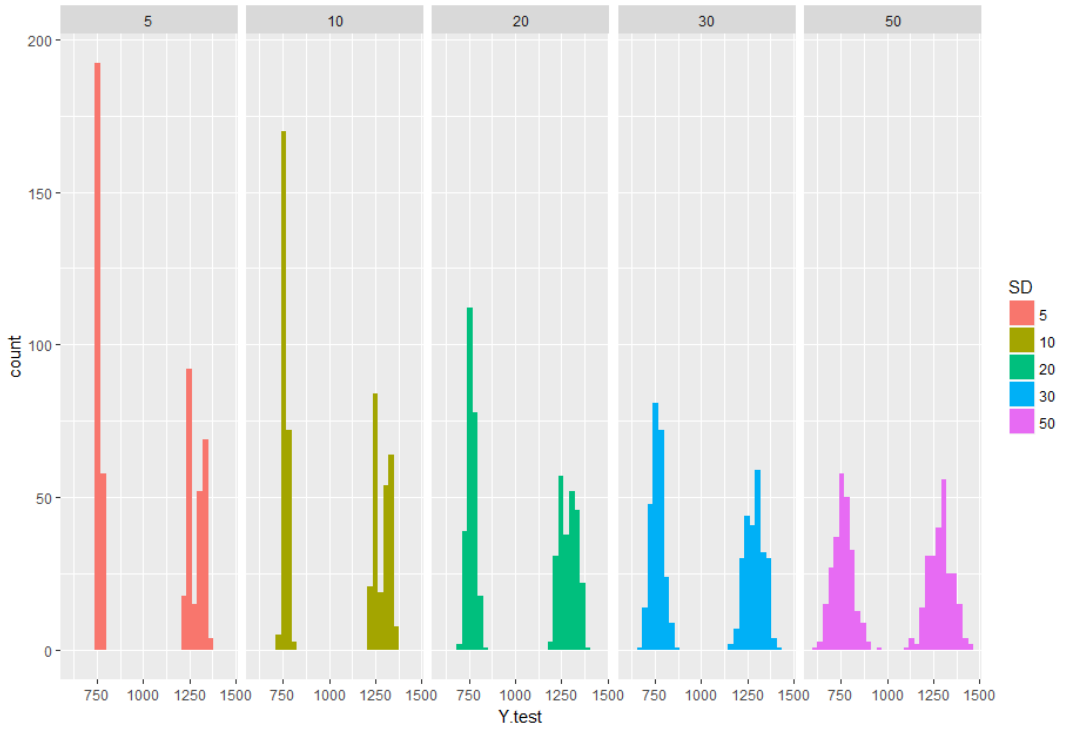


Figure 5.3: Simulation result

SD_E	SD_Y_tr	SD_Y_te	MSE_tr	MSE_te	MAE_tr	MAE_te	MAE_Y_tr	MAE_Y_te	Iter	tau	rank
5	266.73	263.68	7.06	3011.10	2.14	43.61	263.81	261.75	10000	10	18
10	267.12	264.13	79.75	2959.54	7.06	43.06	264.10	262.04	10000	20	6
20	268.13	265.31	79.71	3357.90	7.08	45.96	264.67	262.60	10000	40	7
30	266.87	269.44	301.32	3772.66	13.54	48.92	265.26	263.17	10000	40	3
50	74483.35	73495.91	321.80	5758.41	14.07	60.68	266.42	264.30	10000	40	6

5.2 Neuroscience Application

In this second experiment, we illustrate how SVT can be applied to predict the composite creativity index (CCI) based on brain connectomes. The dataset contain two parts: (1) numeric CCI of 109 volunteers, scored using the Consensual Assesment Technique (Amabile (1983)) and (2) a collection of adjacency matrices of size 70−by−70, each of which is converted from a volunteer's Multimodal Magnetic Resonance Imaging (MRI). Note that the matrices are

symmetric, sparse, and entries are either 1 or 0. Techniques of transforming from MRI to graphs are discussed in Brant-Zawadzki, Gillan, and Nitz, 1992, Desikan et al., 2006, and Kiar et al., 2016. In the past, most studies focus on finding and testing which brain regions significantly affect CCI (Arden et al., 2010); in addition, Wang, Vogelstein, and Priebe, 2017 provides a different perspective: extract important loadings by jointly embedding all graphs and regress CCI on the loadings. Here we adopt a brand new approach by directly constructing the trace regression model (4.3) to predict CCI. The procedure is summarized below. First, we obtain a low rank approximation of each adjacency matrix ($\hat{\mathbf{A}}_i$ in model (4.3)) by selecting the three largest eigenvalue of $\mathbf{A}_i$. This allows us to denoise the input. Then, we manually split the dataset into 10 folds and each time assign one of the fold as the testing sample and the rest the training. We apply the SVT algorithm on the training sample to estimate the unknown coefficient matrix in model (4.3) by treating the $\hat{\mathbf{A}}_i$'s as the independent variables and CCI as the response variable.That is,

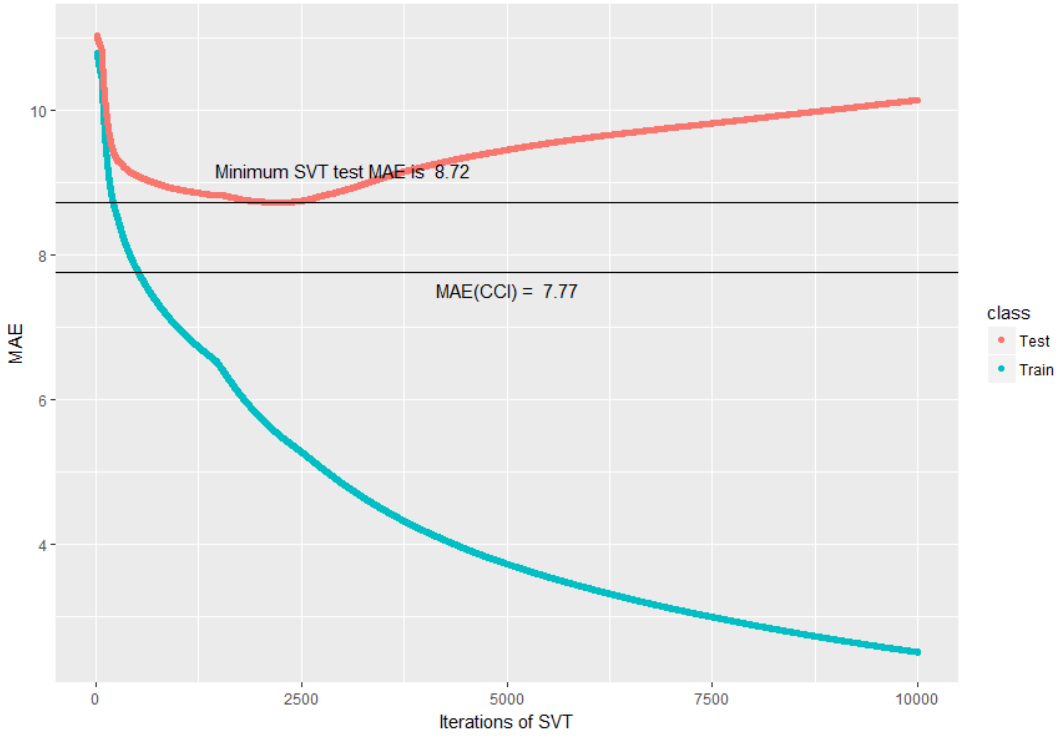
$$CCI_i^{\text{train}} \sim \text{trace}(\hat{\mathbf{A}}_i^{\text{train}} \cdot \Theta) + \epsilon_i$$

Finally, we predict CCI of the testing sample using $\hat{\Theta}$ and compare the prediction with the true values.

Note that since we observe the overfitting effect from the previous simulation study, we record the performance of SVT on both the training and the testing sample over each iteration of the SVT algorithm.

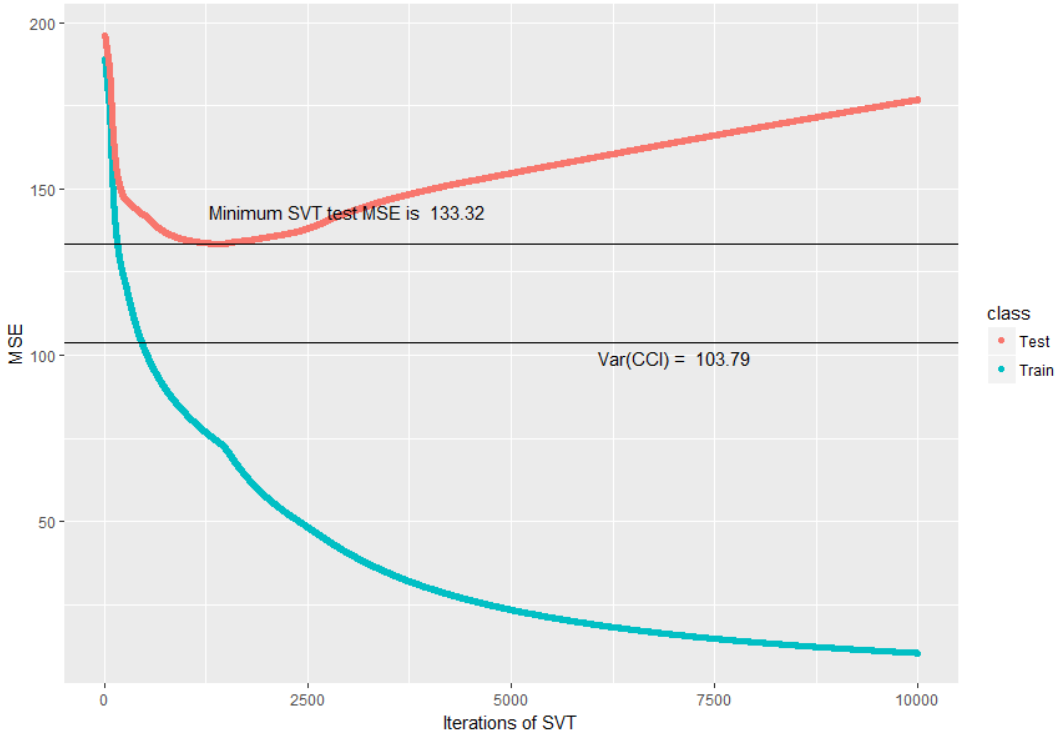
From Figure 5.4 and Figure 5.5, we see a clear overfitting phenomenon in both MAE and MSE: as the iteration increases, training MSE and MAE

Figure 5.4: MAE of SVT



keep decreasing while testing MSE and MAE first decline and then rise. The optimal MSE and MAE both exceed the corresponding MAE and variance of CCI, suggesting SVT does not outperform the mean of CCI in terms of prediction accuracy. However, the discrepancy of the results between SVT and the mean of CCI is not that large. On the other hand, the rank of $\hat{\Theta}$ across the whole range of iterations are small (usually from 3 to 5). Together, these imply that although SVT is not better than the mean of CCI in this dataset, it can to some degree select the important loadings from the graphs and therefore improve interpretability of the model.

Figure 5.5: MSE of SVT



References

Amabile, T. M. (1983). "The social psychology of creativity: A componential conceptualization". In: *Journal of personality and social psychology* 45.2.

Brant-Zawadzki, M., G. D. Gillan, and W. R. Nitz (1992). "Mp rage: a three-dimensional, t1-weighted, gradient-echo sequence–initial experience in the brain". In: *Radiology* 182.3, pp. 169–775.

Desikan, R. S., F. SÂť egonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, and B. T. Hyman et al. (2006). "An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest". In: *Neuroimage* 31.3, pp. 968–980.

Kiar, G., W. Gray Roncal, D. Mhembere, E. Bridgeford, R. Burns, and J. Vogelstein (2016). "ndmg: NeurodataâĂŹs mri graphs pipeline". In: *open-source code. [Online]. Available: http://dx.doi.org/10.5281/zenodo.60206*.

Arden, R., R. S. Chavez, R. Grazioplene, and R. E. Jung (2010). "Neuroimaging creativity: a psychometric view". In: *Behavioural brain research* 214.2, pp. 143–156.

Wang, Shangsi, Joshua T. Vogelstein, and Carey E. Priebe (2017). "Joint Embedding of Graphs". In: *arXiv preprint at https://arxiv.org/abs/1703.03862*.

Chapter 6

Discussion and Conclusion

In summary, the novelty of this thesis lies in the combination of (i) the generalization of the trace regression model which has yet well investigated in the context of graph input and (ii) the application of SVT, an algorithm originally proposed for the purpose of matrix completion instead of trace regression estimation. Through simulation study and experiment on real dataset, we demonstrated the practicality of the model and the estimation method; the procedure can thus be applied to other disciplines. Future work could be devoted into further exploration of the estimation method that may potentially produces not just low-rank but ideally sparse estimation of the coefficient matrix, whereby one is able to screen out exactly which vertices (or edges) play the most important role in determining the response. However, the pursuit of sparsity may contradict with the low-rank property (considering the identity matrix which is sparse but of full rank) and therefore awareness should be raised when deciding which method best fits the context of the problem encountered.

Fanwen Zhu

33 Rogers St., Apt 402, Cambridge, MA, 02142 || (443)255-9770 || fzhu2@jhu.edu

EXECUTIVE SUMMARY

- High achiever with consistent passion for interdisciplinary research, excelling at statistical learning and economic analysis
- Particular interest in combining economics as well as finance with psychology and statistical learning

Education

Johns Hopkins University, Baltimore, MD
Master of Science and Engineering in Applied Math and Statistics 8/18

- Concentration: Statistics and Statistical Learning
- Coursework: Measure-theoretic Probability, Data Mining, Time Series Analysis, and Nonlinear Optimization
- Overall GPA: 3.95/4.00
- Master Thesis: "Graph Trace Regression Estimation"
- Thesis description: Develop supervised learning method predicting numerical response variable where explanatory variables are "graphs" with vertices and edges. Apply model to neuroscience dataset predicting individual composite creativity index via brain "graphs"

Johns Hopkins University
Bachelor of Arts in Applied Math and Statistics 5/17

- Second Major: *Economics* ; Minors: *Mathematics* and *Financial Economics*
- Overall GPA: 3.72/4.00; Applied Math and Statistics Major GPA: 3.72; Economics Major GPA: 3.86
- Consecutive Dean's List Awards; Graduated with honors
- Relevant Coursework: Probability, Statistics, Stochastic Calculus for Finance, Econometrics, Intermediate Micro- and Macro-Economics, Financial Markets, Investment Theory, and Corporate Finance
- Senior Term Paper: "China's Dual-Track Economic Reform and Problems in China's Banking System"
- Paper description: Study conflicts between China's planned- and market-tracked economic structure: Their historical and political origins, manifestation in China's banking system, and relation to China's interest rate reform and restructuring of state-owned enterprises

EXPERIENCE

MIT – Sloan School of Management
Research Assistant August 2018-Now

- Conduct empirical econometric analysis in the fields of macroeconomics and financial economics

Fuels Institute – Nationwide Business Case Competition, U.S.
Team Member of 4 Participants Fall 2017

- Awarded honorable mention
- Crafted market analysis and performed NPV-related analysis on project evaluation

Johns Hopkins University – Applied Math and Statistics Department
Teaching Assistant for Applied Statistics and Data Analysis (Graduate level) Fall 2016

- Covered matrix-based linear and generalized linear regression models
- Taught weekly sessions on regression analysis theories

SKILLS

- Language: Fluent in English and Chinese.
- Programming: Python, R, STATA, MATLAB, Latex, and Markdown
- Ballroom dancing: Bronze-level Waltz, Quickstep, Tango, Cha-cha, Rumba and Jive